FFI-RAPPORT

20/02840

Exploring data reuse using a big data infrastructure

Jonas Halvorsen
Bjørn Jervell Hansen

Exploring data reuse using a big data infrastructure

Jonas Halvorsen
Bjørn Jervell Hansen

Norwegian Defence Research Establishment (FFI) 5 November 2020

Keywords
Stordata
Databehandling
Informasjonsinfrastruktur
Informasjonsintegrasjon

FFI report
20/02840

Project number
1430

Electronic ISBN
978-82-464-3304-2

Approvers
Jan Erik Voldhaug, *Director of Research*
Trude Hafsøe Bloebaum, *Research Manager*

The document is electronically approved and therefore has no handwritten signature.

Summary

Making good military decisions requires a high level of situational awareness, and building this situational awareness is improved by access to as much relevant information as possible. This information can arrive to a decision maker via many different avenues, one of which is the reuse of information already collected or prepared for other purposes.

Data reuse is acknowledged as an important ingredient in the process for a military organization to fulfill their information needs by both NATO and the Norwegian Armed Forces as they the last 15 years have sought to turn their data strategies from the traditional need-to-know to the more open responsibility-to-share paradigm.

Ubiquitous information sharing and reuse have, however, certain prerequisites in order for it to happen. For example, the sharer of data must have trust that only authorized users will have access to it. The potential user, on the other hand, must be able to determine the provenance and reliability of the data, and whether or not it is in a suitable format, before eventual use.

This report documents a technical experiment setting out to explore whether it is feasible to build a big data infrastructure with the appropriate requirements to make it suitable for data reuse in the military domain using open source components. The exploration is supported by an experimental setup that expands on a previously explored big data infrastructure based on open-source components, extending it with suitable components for facilitating data reuse. Specifically, the two lines of inquiry explored in this report are

1. Simplifying the re-purposing and joining of data sets by publishing data as *linked data*, which is a structured representation that makes it easy to interlink with other data.
2. Utilizing lineage-based data governance for provenance tracking and fine-grained access control in a big data ecosystem that is comprised of many different components.

The technical exploration is performed against a fictitious backdrop of real-time news analysis, where a team of analysts keeps track of events in a region in support of an on-going military operation. This case requires merging of information from real-time news streams together with static background knowledge. The technical infrastructure is laid out and explained from a conceptual level, including brief introductions to the components used. Key features, as well as how they address the outlined issues with respect to data reuse, are explained and highlighted through the use of the underlying news analysis case.

As identified in a previous FFI report, there is no one generic big data infrastructure that fits all; the choice of big data components in an infrastructure is very much dictated by the case and problem at hand, and the setup explored in this report, which was crafted for a specific case, is no exception to this. The main contribution of this report is that it provides the reader with an example of how today's open-source, off-the-shelf big data technologies from the civilian sector can be utilized in the military domain to facilitate data reuse, governance and fine-grained access control. The results are thus supporting evidence of the feasibility of building such an infrastructure, and can be of utility for personnel considering different architectural approaches for dealing with information management in a military setting.

Sammendrag

Å ta gode militære avgjørelser krever et høyt nivå av situasjonsbevissthet, noe som kan forbedres ved å ha mest mulig relevant informasjon tilgjengelig. Denne informasjonen kan nå en beslutningstaker via mange forskjellige veier, hvorav gjenbruk av informasjon som allerede er samlet inn eller forberedt for andre formål står sentralt.

Gjenbruk av data er anerkjent som en viktig ingrediens i prosessen med å tilfredsstille informasjonsbehovet i militære organisasjoner: Både NATO og Forsvaret har i de siste 15 årene fokusert mindre på det tradisjonelle *need-to-know*-regimet og mer på en mer åpen *responsibility-to-share*-strategi. Imidlertid er det visse forutsetninger som må på plass for at omfattende informasjonsdeling og gjenbruk skal kunne skje. En datatilbyder vil kunne kreve forsikringer om at data som deles kun blir tilgjengelig for autoriserte brukere, mens en potensiell bruker vil kunne ønske å ettergå dataenes opphav for å vurdere om kvaliteten, påliteligheten og formatet er adekvat for ønsket bruk.

Rapporten beskriver et eksperiment der formålet var å undersøke hvorvidt det er gjennomførbart å bygge en stordatainfrastruktur med egenskaper som gjør det mulig å gjenbruke data kontrollert i en militær kontekst. Eksperimentet besto i å ta utgangspunkt i en tidligere dokumentert stordatainfrastruktur bestående av åpen-kildekode-komponenter og utvide denne infrastrukturen for å fasilitere datagjenbruk. Konkret er følgende to spor utforsket:

1. Publisere data som *lenkede data*, en strukturert datarepresentasjon som gjør det lett å knytte data sammen med andre data, for å forenkle gjenbruk og integrasjon av forskjelligartede datasett.
2. Utnytte og avlede opphavshistorikk for å styre data og tilby provenansstyring og finmasket aksesskontroll i et stordata-økosystem med mange forskjellige komponenter.

Eksperimentet ble utført ved hjelp av en oppdiktet case på nyhetsanalyse i sanntid, der et tenkt team av analytikere holder oversikt over hendelser i en region til støtte for en pågående militæroperasjon. Dette tilfellet krever at informasjon fra sanntids nyhetsstrømmer blir slått sammen med statisk bakgrunnskunnskap. Rapporten beskriver den tekniske infrastrukturen som ble satt opp for å løse casen på et konseptuelt nivå, og går igjennom hvordan de behandler de skisserte problemene med hensyn til gjenbruk av data.

Som identifisert i en tidligere FFI-rapport, finnes det ikke en generisk stordatainfrastruktur som passer alle brukstilfeller: Hvilke komponenter infrastrukturen består av bør dikteres av de karakteristiske trekkene ved det problemet som skal løses. Infrastrukturen som blir utforsket i denne rapporten, er heller ikke noe unntak ettersom den ble satt sammen for å løse en bestemt nyhetsanalysecase. Rapportens viktigste bidrag er å gi et eksempel på hvordan man kan utnytte stordatateknologier (basert på åpen kildekode) fra sivil sektor i det militære domenet for å legge til rette for økt, kontrollert datagjenbruk Dette støtter hypotesen om at dette kan gjennomføres, og rapporten kan være nyttig informasjonsgrunnlag når man vurderer forskjellige tilnærminger for å håndtere informasjonsforvaltning i en militær kontekst.

Contents

1 Introduction

Making good military decisions requires a high level of situational awareness, and building this situational awareness is improved by access to as much relevant information as possible. This information can arrive to a decision maker via many different avenues, one of which is the reuse of information already collected or prepared for other purposes.

The importance of data reuse is embraced by NATO, as seen by their decision to turn their data strategy from the familiar need-to-know, where the emphasis is on preventing information reaching the wrong recipients, to the more data reuse oriented responsibility-to-share, where the emphasis is on making sure information reaches those who need it (NATO 2008).

This is as relevant to the Norwegian Armed Forces as to NATO, as can be witnessed by the highlighting of responsibility-to-share as an important principle to promote transparency as a means to enhance situational awareness and reduce the possibilities of mistrust, suspicion, and discord in military operations (Forsvaret 2019).

In a previous activity (Stolpe et al. 2020) the focus was on developing and demonstrating a big data infrastructure for multi-modal stream processing. The current exploration aims to extend that set-up with features that address the general issue of data reuse in a big data infrastructure.

Our starting point is the question whether it is feasible to build a big data infrastructure with the appropriate requirements to make it suitable for data reuse in the military domain using open source components. Our adopted methodology to provide answers to this question is use-case driven and explorative: We have chosen a use case concerning real-time streaming global news events integrated with structured background knowledge, putting us in a position to explore the issue using real streaming data. The chosen data also has potential military relevance, as news can be used to build situational awareness about, among other things, the society in which a military operation is taking place. Further, we have explored this in a practical way building an actual big data infrastructure with common open-source components and put this infrastructure to the test with the use case streaming data.

Specifically, the two lines of inquiry are

1. utilizing *Linked Data* principles (Heath & Bizer 2011), that is, using a graph-based data representation and Web-based unique identifiers, to simplify the repurposing and joining of data sets, and
2. utilizing lineage-based data governance for provenance tracking and fine-grained access control in a big data ecosystem that is comprised of many different components.

We note that, for the purpose of this report, data reuse refers to using existing data sets either the way the data creator intended it for (e.g. same task or domain) or as novel, unintended use (for a different task). The latter is often referred to as repurposing, hence a more specific term, and we shall only use that specific term when we wish to specifically highlight unintended use.

The report is primarily written for a technical audience, and is specifically aimed at system architects and others that work with technologies that deal with the issue of information management. As such, the main contribution of this report is that it provides the reader an example of how today's

commonly used big data technologies from the civilian sector can be utilized in the military domain to facilitate data reuse. This can be of utility when considering different architectural approaches for dealing with information management in a military setting.

This report is structured as follows: Chapter 2 provides a brief overview of data reuse, why it is important, and what needs to be in place for it to happen. Chapter 3 takes this further, by looking into what specific requirements incur when assuming a big data infrastructure with several distinct, distributed components. In Chapter 4, we describe a use case for the exploration. A realistic use case is beneficial for both distilling realistic functional and technical requirements for a big data infrastructure, as well as providing a frame with respect to choosing suitable components for the task at hand. Followingly, in Chapter 5 we describe a concrete infrastructure that suits the case at hand, together with descriptions of the chosen components. Chapter 6 outlines a use of the infrastructure where information from news articles enhanced with static data are reused for analysis purposes, while Chapter 7 explains how the necessary data security and use restrictions are upheld in the infrastructure. Finally, Chapter 8 provides a summary and a short discussion regarding the benefits and tradeoffs of the experimental setup explored.

2 Facilitating and Enabling Data Reuse

The reuse of data has many obvious potential advantages to organizations needing data to inform decisions. Time, money, and resources can be saved as data is collected once and exploited several times, data can be more readily available as it has already been collected, and the amount of data available for analysis can increase, to name a few. This is as true for military organizations as for any other organization.

When data is being reused, there are two parties involved, namely the sharer of data and the potential user, and they respectively have differing requirements that need to be addressed adequately in order for data reuse to happen.

A user looking for data to cover his or her information needs, needs to be able to find it, trust it, and finally use it. That is, the user will have to discover that the information artifact exists, is in a suitable format and data model that s/he can utilize, and is produced by a trusted actor and according to the user's quality requirements.

From the point of the data sharer, trust is the central issue. In order for the data sharers to be willing to share their valuable data, they must be able to trust that only authorized users can access the data, that it is used according to specific restrictions, and that any derived information products continue to adhere to (at least) the restrictions set on the source data. That is, access restrictions follow data lineage.

The overview presented on the issues related to data reuse is quite cursory. For a more detailed analysis, as well as best practices and strategies, we would direct the reader towards the FAIR Data Principles for scientific data management and stewardship (Wilkinson et al. 2016), as well as the "*Veileder for tilgjengeliggjøring av åpne data*" (Norwegian Digitalisation Agency 2019):

- The FAIR data principles[1] provide rule-of-thumb guidelines on how to make data Findable, Accessible, Interoperable, and Reusable, all necessary features in enabling data reuse. These principles have been widely adopted, and were given endorsement by the G20 countries in 2016 for ensuring access to publicly funded research results.
- The "*Veileder for tilgjengeliggjøring av åpne data*"[2] by the Norwegian Digitalisation Agency (DIFI), and the more generic "*Retningslinjer ved tilgjengeliggjøring av offentlige data*"[3] by the Norwegian Ministry of Local Government and Modernisation, provide best practices to follow in order to make public data accessible for reuse. It outlines, amongst other things, the use of strong URIs as resource identifiers, and machine readable formats.

The work presented in this report is highly aligned with both the FAIR principles and the data publishing guidelines provided by DIFI, although it does not follow these fully to the point. Moreover, we have chosen to use *graphs* as the central data representation formalism when facilitating for data reuse. Using graphs for data and knowledge representation has a long history, see e.g. Hogan et al. (2020), and by representing data this way, reuse is facilitated by the inherent flexibility offered by the graph format.

[1] https://www.go-fair.org/fair-principles/

[2] https://doc.difi.no/data/veileder-apne-data/

[3] https://www.regjeringen.no/no/dokumenter/retningslinjer-ved-tilgjengeliggjoring-av-offentlige-data/

With this as a backdrop, we refine the outlined exploration lines with candidate open-source solutions that are suited for our existing infrastructure, and the case at hand, as follows:

1. Simplify repurposing and joining of data sets by using the *Linked Data* (Heath & Bizer 2011) graph representation scheme, where data elements with unique identifiers are linked together in graphs, represented using the Resource Description Framework (RDF) (Carroll & Klyne 2004) abstract format and queryable using the RDF graph query language SPARQL (Seaborne & Harris 2013).
2. Lineage-based data governance and access control, over a big data ecosystem with many different components, facilitated through the combination of Apache Atlas (Apache Software Foundation 2020*a*), a governance and metadata catalogue, and Apache Ranger (Apache Software Foundation 2020*d*), a decentralized policy framework.

3 Conditions and Requirements for Data Reuse in a Big Data Infrastructure

As mentioned earlier, a previous experiment (Stolpe et al. 2020) focused on developing and demonstrating a big data infrastructure for multi-modal stream processing. The current experiment aims to extend this set-up with essential features that address the general issue of data reuse and repurposing in a big data infrastructure.

In order to identify the proper infrastructure requirements for this experiment, we started from the requirements outlined in the previously mentioned experiment (Stolpe et al. 2020) and sought to add requirements needed to cover the data reuse topic. These former requirements are:

- Timeliness: The infrastructure is able to make data available in real or near real time.
- Scalability: The infrastructure is able to scale automatically in the event of increased data load.
- Parallelizability: The infrastructure supports parallelizable algorithms.
- Loose coupling: The infrastructure is able to handle new data producers and -consumers on the fly.
- Accountability: The infrastructure is able to keep track of the data to ensure that it is verifiable and retraceable.
- Fault tolerance: The infrastructure is able to continue operating properly even in the event of the failure of some of its components.
- Reliability: The infrastructure ensures that the data is correct with respect to temporal ordering.

These requirements do not, however, properly cover the data reuse case described in Chapter 2. As mentioned there, this can be seen from two different perspectives: The perspective of the data user, and the perspective of the data provider.

To start with the data provider, the main issue is trust in the infrastructure. The infrastructure must have mechanisms that ensure that the data is protected in accordance with the data provider's requirements, mechanisms which we in the following will label access management.

For the data user, the main issues are a) being able to find trustable data of an adequate quality in a format s/he is able to make use of, and b) having confidence that the data's integrity has not been compromised. For the former, the infrastructure should provide functionality that allows the user to audit where data originated from as well as any processing steps it has undergone along the way, which are essential for determining trustworthyness and quality of the data product. The latter is ensured by the infrastructure providing proper access management mechanisms.

To summarize, the big data infrastructure should also meet the following requirements:

- Access management: The infrastructure makes sure that access to both original and derived data is restricted in accordance with the appropriate policies.
- Reuseability: The infrastructure provides extensive auditing functionality as well as generic, standardized interfaces that promote data reuse.
- Refindability: The infrastructure facilitates that users can find the information they need.

4 Use Case: Real-Time News Analysis

The experimental case study is performed with the following use case in mind: An analyst team keeps track of events in a region in support of an on-going military operation. The analysts in question need to use all available sources to build their situational awareness, making it crucial that they are able to easily reuse whatever information they can get their hands on.

During the operation, a new source of event information is brought to their attention: News events from the Global Database of Events, Language, and Tone, aka. the GDELT project[4] (Leetaru & Schrodt 2013). They now need to feed this information into their existing infrastructure in order to reuse it. Further, they want to enrich the GDELT data with static background data on people and places from the online, structured source DBPedia (Auer et al. 2007).

The two new information sources are described further below.

4.1 Data source: GDELT

The GDELT project is an initiative to collect and catalog news events, and provide it to the world for analysis reuse, free of charge. Examples of how the GDELT data has been used, includes analysis to support the claim that the Arab Spring in 2011 sparked a wave of global protests (Leetharu 2014), analysis of news media coverage of refugees (Boudemagh & Moise 2017), and predicting social unrest events (Qiao et al. 2017).

In addition to keeping the news event data itself, GDELT performs extraction of actors, locations, and themes and accumulates these in a graph named the GDELT Global Knowledge Graph (GKG). Both the news event data and the global knowledge graph are made available every 15 minutes in comma-separated values (CSV) files, see an example in Figure 4.1.

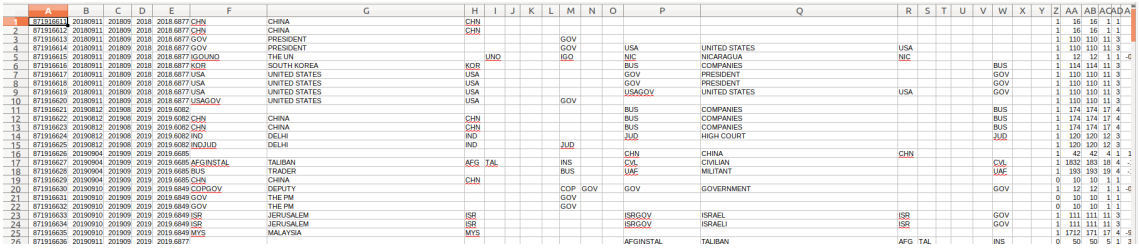


Figure 4.1 An example of data from GDELT.

The GDELT data is centered round the following data types:

- Events: The actors involved in the event, and a link to the corresponding news article.
- Mentions: All news articles where an event is mentioned.
- GKG: GKG-elements (actors, locations, themes) related to a specific news article.

[4]https://www.gdeltproject.org/

4.2 Data source: DBPedia

DBPedia is a project that extracts structured information from Wikipedia, the large crowd-sourced, web-based encyclopedia. The project was initiated by University of Mannheim, Leipzig University, and OpenLink Software in 2007, and it maintains and publishes a graph-based database of the extracted entities in the Resource Description Framework (RDF) format (Carroll & Klyne 2004).

DBPedia is an important part of the web-wide Linked Data initiative that was initiated by Tim Berners-Lee in 2006. A central principle in Linked Data is that all entities should be represented by URIs (Uniform Resource Identifiers), and thus be uniquely identifiable. Another Linked Data principle is that these URIs should be HTTP based, and thus de-referenceable (i.e. when you enter the URI in a web browser, you will reach a representation of the entity). These principles make DBPedia information easily reusable: All you need to get information about an entity is its HTTP-based identifier (Heath & Bizer 2011).

In Figure 4.2, an example of the DBPedia representation of the Wikipedia information on the Norwegian Defence Research Establishment (FFI) is shown.

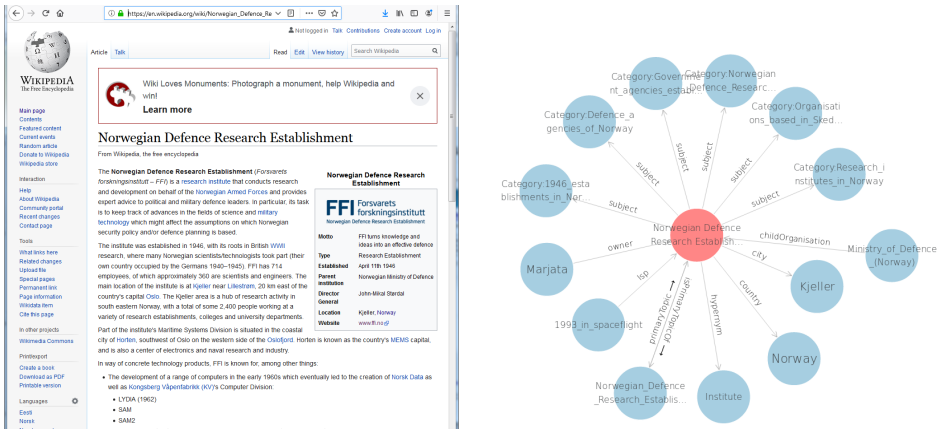


Figure 4.2 Information about FFI in Wikipedia (left) and DBPedia (right).

5 A Big Data Infrastructure for Data Reuse in Real-Time Event Processing

In order for our analyst team to be able to reuse the information outlined in Chapter 4, they need an appropriate infrastructure. In this chapter we will present a big data infrastructure designed for this use case according to the requirements presented in Chapter 3, and will also present the different components that make up the infrastructure.

The chosen infrastructure components and their interaction are shown in Figure 5.1, and the interaction between the components can be outlined thus: GDELT data (events, gkg, mentions) are collected as CSV files, and preprocessed into RDF before being sent to the publish/subscribe message bus (Kafka), labelled appropriately. All further treatment of these data elements is done by components collecting the data from Kafka according to the labels:

- Data is enriched with correct geo attributes by a stream query component (the upper node labelled C-SPARQL in the figure),
- analytics information is extracted from the data stream by another stream query component (the lower node labelled C-SPARQL in the figure),
- a custom processor monitors the data streams for DBPedia links in order to query more information about them from DBPedia, and
- data is continuously sent to the a graph database (GraphDB) for later to be displayed to the analysts.

The data flow between these components was set up using Apache NiFi. NiFi was also configured to report all processing automatically, allowing for the processing lineage for each data element to be preserved (in Atlas).

Finally, access control to the infrastructure was set up using Apache Ranger, in cooperation with Apache Atlas.

5.1 Infrastructure component: Apache NiFi

Apache NiFi is a software project from the Apache Software Foundation designed to automate the flow of data between software systems (Apache Software Foundation 2020*c*). It is based on the *NiagaraFiles* software previously developed by the US National Security Agency (NSA), and was open sourced as a part of NSA's technology transfer program in 2014.

NiFi is essentially a visual application development environment managing adapters to different data sources, and allows assembling pipelines for data routing, transformation and system mediation logic. It has a comprehensive library of adapters to the most commonly used IT components, and is thus well suited to situations where new information sources have to be taken into account. NiFi is designed for scalability and fault tolerance, and is easily set up to operate on machine clusters.

NiFi has recently been added as a core component to the Cloudera Data Platform (Cloudera 2019).

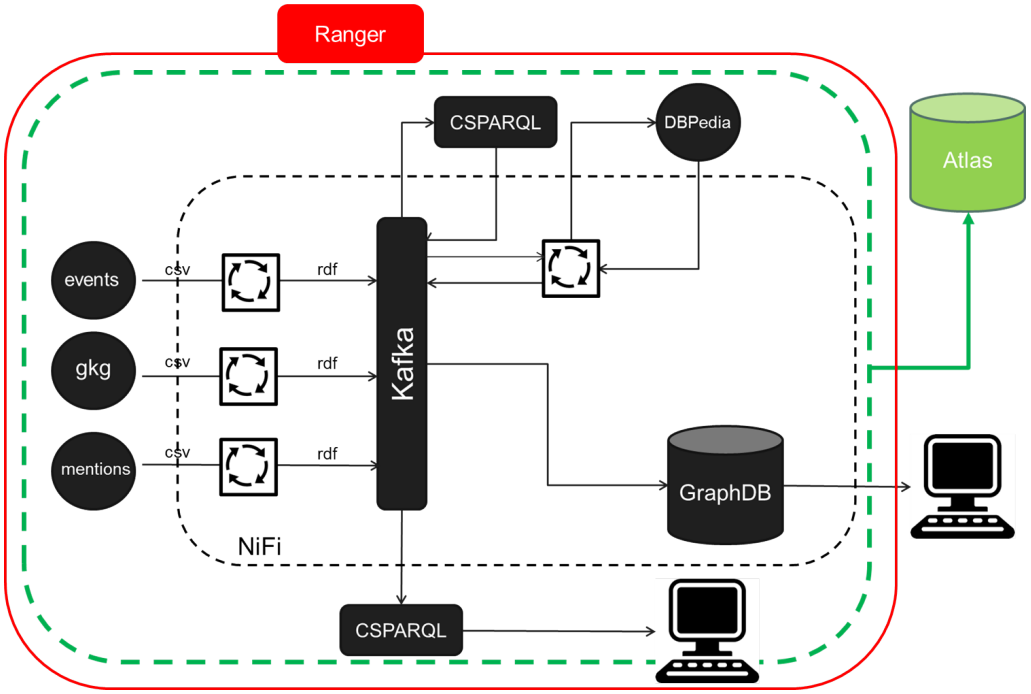


Figure 5.1 The chosen big data infrastructure.

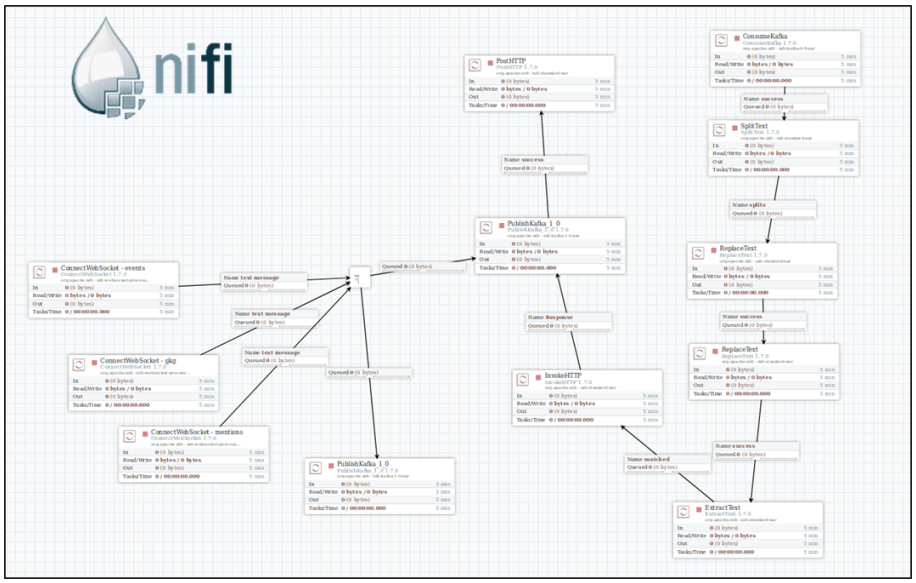


Figure 5.2 A NiFi information flow.

Our experience with NiFi in this experiment backs up our previous experience: It is easy to install, and its user interface makes it relatively straight forward to set up the necessary information flows as the one shown in Figure 5.2.

5.2 Infrastructure component: Apache Kafka

Apache Kafka is a distributed stream processing software platform originally developed by LinkedIn but donated to the Apache Software Foundation and open sourced in 2011 (Apache Software Foundation 2020*b*). Kafka is capable of handling high velocity, high throughput data in real time and of scaling up as workloads increase.

Kakfa is a scalable publish/subscribe messaging system designed as a distributed transaction log. It distributes and stores streams of messages organized in different labels, or *topics*, distributing the messages to all consumers listening to the appropriate topic and keeping all the transported messages in logs. It also provides guarantees that not only will all messages be delivered, even in the face of failure, they will be delivered exactly once. Furthermore, Kafka can be set up to guarantee that the temporal ordering of messages will be kept.

As is the case with NiFi, Kafka has also recently been added to the Cloudera data platform (Cloudera 2019). Even before this, Kafka could be considered a de facto big data solution standard for asynchronous messaging, with LinkedIn, Spotify, and Netflix, as well as the Norwegian power system operator Statnett, as noteable users.

As in our previous explorations, Kafka appears as a robust pub/sub component providing the overall system with loose coupling between the components, and well suited to streaming data tasks. It is also easy to install and configure.

5.3 Infrastructure component: C-SPARQL

C-SPARQL (Countinuous SPARQL) is an extension of the SPARQL Protocol and RDF Query Language (SPARQL) (Seaborne & Harris 2013), making it suitable for querying streaming graph data with continuous queries.

The following extensions make this possible (Barbieri et al. 2010):

- RDF Stream Data type, specifying a new data type that defines streaming RDF data.
- Window semantics, allowing time windows to be defined over which the queries can be evaluated.
- Stream registration, allowing new RDF streams to be defined from the output of queries.
- Query registration, allowing the definition of queries that are periodically evaluated.
- Multiple streams, allowing triples from more than one RDF stream to be queried.
- Timestamp function, keeping track of the timestamp for each stream element.

In this particular infrastructure, a C-SPARQL query engine was mainly used to continuously query the GDELT stream for events and summarize them per country. The benefit of this operation is

explained in Chapter 6.

5.4 Infrastructure component: Apache Atlas

Apache Atlas is an open-source framework for data governance and metadata management over assets stored in big data ecosystems. Its metadata core is built around a graph-based store, JanusGraph, and the framework was originally designed for Hadoop-based environments, but has grown to support and cater for a wider variety of components. In addition to the Hadoop stack, this includes components such as Kafka and NiFi (see Sections 5.2 and 5.1). Like those two compoents, Atlas is a core component to the Cloudera Data Platform (Cloudera 2019).

Notable features that the system provides is security classification tagging of assets and lineage-based metadata propagation, down to column-level granularity, in addition to more traditional catalog functionality such as metadata curation, business term tagging and search facilities. These features can be utilized by Apache Ranger for fine-grained access control, which will be further explained in Section 5.5.

Many big data components have features that automatically generate and report in metadata to Atlas when artifacts or processes are created in the host systems (e.g. Hive tables, Kafka topics, or even NiFi flows), through the use of Atlas's REST interface or a Kafka ingest topic. In the case of NiFi flows, the metadata contributed will not only include individual artifacts, but also the information that describes the lineage relationship between them.

For our use, Atlas acts as a metadata catalogue over artifacts that exist in the ecosystem, and together with Apache Ranger, provides a holistic governance platform.

5.5 Infrastructure component: Apache Ranger

Apache Ranger is an open-source framework for managing, enforcing and auditing data access in big data infrastructures and data lakes. It affords a single solution for authoring and enforcing access policies over a variety of different components, providing fine-grained role- and attribute-based access control methods (RBAC/ABAC respectively) and data access logging.

What makes Ranger uniquely suited for the task at hand is that

1. it supports a wide range of big data components out-of-the-box (e.g. Kafka, NiFi),
2. it provides a unified platform for managing policy rules over these disparate systems (see Figure 5.3),
3. policy decision and enforcement are performed locally on the participating components rather than being centralized (thus reducing single points of failure),
4. fine-grained access restrictions can be set as low as at the row- and column-level,
5. it provides detailed audit logging features, and finally,
6. it can utilize security classification tags from Apache Atlas in policy rules.

Ranger affords a unified, centralized platform for policy authoring, management, and auditing

over disparate components, but delegates the actual enforcement to the component level through access control plugins utilizing common Ranger libraries. That is, Ranger-compliant components regularly fetch policies from the policy manager and submit access requests to its audit log, but the enforcement is done locally (i.e. in Kafka or NiFi itself). This means that should the Ranger policy manager go down, the participating components are still able to work isolated and apply the stringent access control as usual. Like NiFi, Kafka, and Atlas, Ranger is a core component to the Cloudera Data Platform (Cloudera 2019).

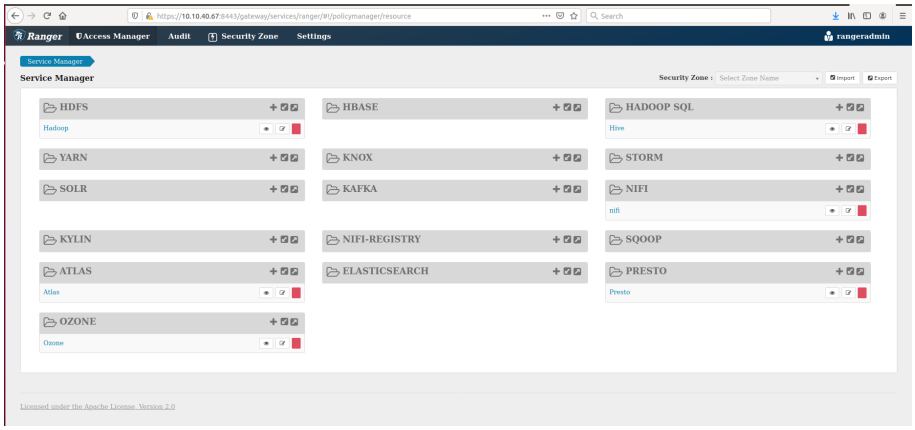


Figure 5.3 The Ranger administrative interface.

5.6 Infrastructure component: GraphDB

As described in (Stolpe et al. 2018), graph databases are important components in big data systems as graphs are versatile and flexible structures well suited to handling integration of different types of data. For this experiment, GraphDB was included to provide the infrastructure with this characteristic.

GraphDB is a graph database designed to hold RDF graphs. It is developed by the company Ontotext[5], and is used by several large companies worldwide, including BBC, Elsevier, Fujitsu, and Financial Times. In this case, it was chosen due to its ease of use and the fact that it is offered in a free version. An example of its user interface is shown in Figure 5.4.

5.7 Requirements revisited

Put together, the components described in this chapter, fulfills the requirements presented in Chapter 3:

- *Scalability* is an inherent feature of Kafka, NiFi, Atlas, Ranger, and GraphDB.
- *Loose coupling* is provided by Kafka in essence being a publish/subscribe messaging system that allows other components to interact in an asynchronous fashion.

[5]https://www.ontotext.com/

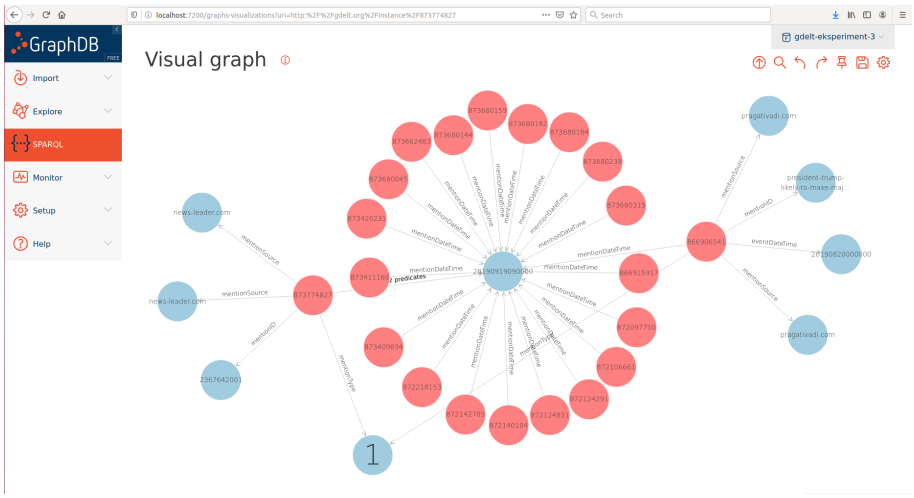


Figure 5.4 The graph database GraphDB.

- *Accountablity* is provided by Kafka, as it stores all data being sent through it in logs that can be examined after the fact to examine the data flow.
- *Reliability* is provided by Kafka's guarantee that messages will be delivered, and that they will be processed exactly once.
- *Access management* is supported by Apache Ranger and Apache Atlas in tandem. Access policies can be established and enforced through the use of Ranger, and Atlas provides the opportunity to manage tags which can be used to implement fine-grained access control policies in Ranger.
- *Reuseability* is enhanced by the both lineage-based governance features in Apache Atlas and the usage of Linked Data as the pervasive data representation.

Timeliness can, in this context, be seen as a result of the inherent scalability of Kafka and NiFi, while the *fault tolerance*, *parallelizability*, and *refindability* were decided to be outside the scope of this exploration due to resource constraints.

6 Data Reuse: Using Linked Data to Enhance News Articles

With our team of analysts now equipped with a suitable infrastructure, they can turn their efforts to the task at hand: Integrating the new information presented in Chapter 4 and reuse this information in order to enhance their situational awareness.

In the following we will use the fictitious team of analysts and their investigation as a common thread, and tell this example of reuse of data in graph form as seen through their eyes.

The first step in this particular investigation was to get a global view of the new data in order to determine where it makes sense to start digging. In order to achieve this, a custom made viewer that could show the number of events by country as reported by GDELT was created. The viewer, and the accompanying infrastructure, was implemented utilizing C-SPARQL (see Section 5.3) to collect and organize the event data. Further details on this implementation can be found in Johannessen (2019).

Figure 6.1 shows an example of a result using this viewer where the number of events per country are shown, as well as a representation of where these events took place along with a their respective conflict intensity according to the Goldstein scale (Goldstein 1992). The intensity is color coded from green (low intensity), through yellow (medium), to red (high intensity).

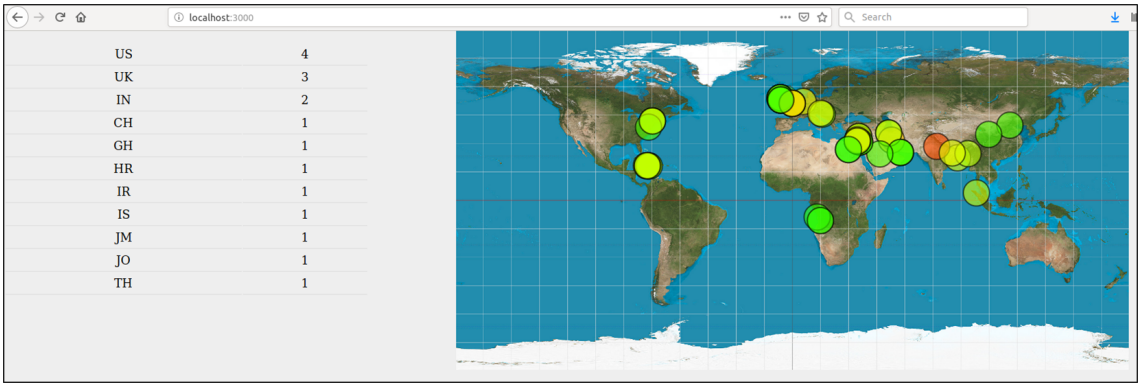


Figure 6.1 Global view of events and their conflict intensity.

Using this overview, the team of analysts could get a quick overview of events in their area of interest. As the data here was represented in a graph, it readily lends itself to drill-down navigation (see e.g. Herman et al. (2000)) where a user can navigate the data by following the nodes and edges of the graph. In this exploration this was performed with GDELT data collected at September 19 2019 enriched by DBPedia data.

At this point, a pipeline was prepared using Apache NiFi (see Section 5.1) where GDELT data was collected as a graph, the data was enriched with information regarding people and places from DBPedia, and the resulting graph was put in the GraphDB database.

The inclusion of DBPedia data was a straight forward integration, helped by the fact that two graphs can be integrated by simply putting them together, and further by the fact that in RDF graphs all

nodes have unique identifiers.

Step 1: Investigating an event From previous investigations, the analyst team had an indication that corruption was a topic of particular interest. Thus they wanted to explore the newly assembled information with this in mind. Based on leads from previous analysis, the investigation started on the event represented by the node *872185997* (see Figure 6.2, red node). Expanding the links from this node resulted in the graph shown in the figure. The figure also shows the (censored) news article on which the event is based: An judiciary appointment to the AP judicial commission.

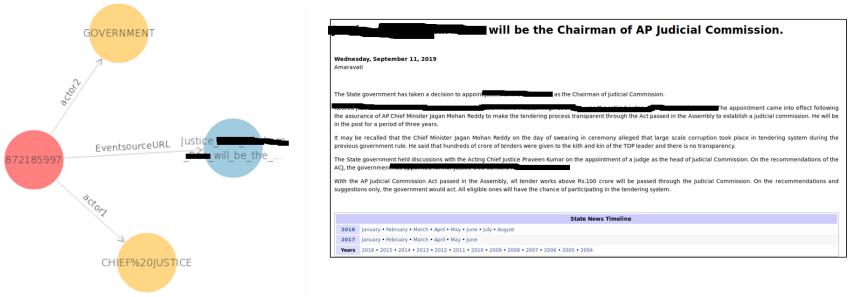


Figure 6.2 Analysis step 1: Expanding an event node.

Step 2: Navigating the links In Figure 6.3 the analyst has traversed the graph by following the link *EventsourceURL*, and expanding the destination node. This reveals more information related to the event of interest: Three more events related to the original event (nodes *872185757* (green), *872185997*(green), and *872185757* (red)), as well as a GDELT Global Knowledge Graph (GKG) node (*20190912083000-730*, see Section 4.1 for a short introduction to GKG).

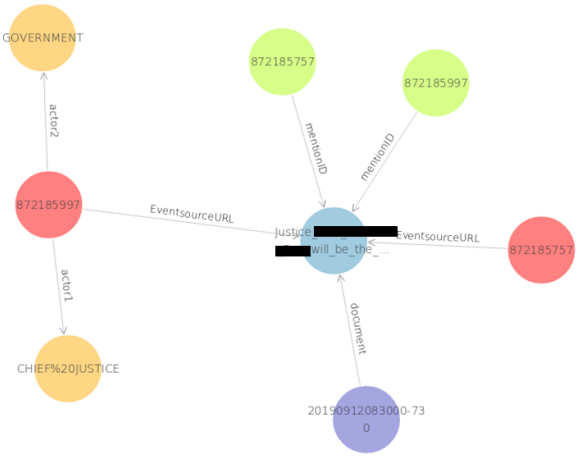


Figure 6.3 Analysis step 2: Traversing the data graph to consider more information.

Step 3: Expanding the graph further The next node of interest to be expanded, is the GKG node *20190912083000-730*. The resulting graph, shown in Figure 6.4, reveals plenty of new

information related to our event: Several nodes representing different themes (e.g. *APPOINTMENT*, *CORRUPTION*) and some nodes representing other people.

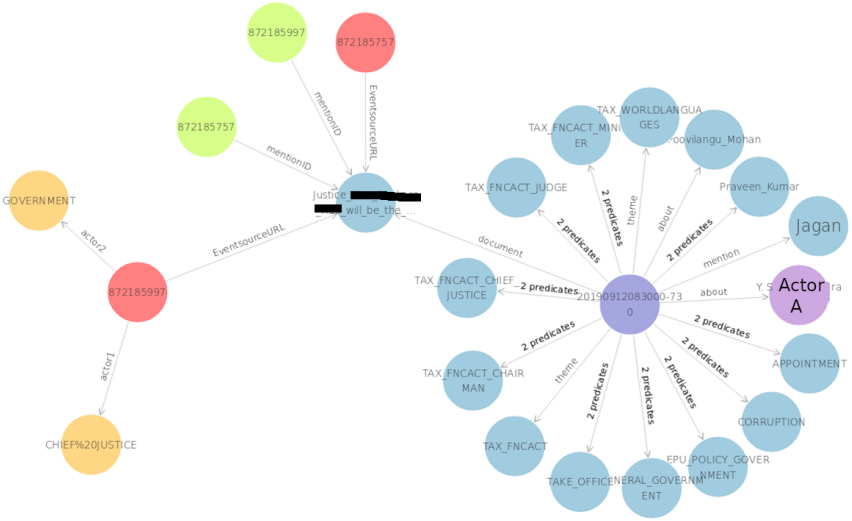


Figure 6.4 Analysis step 3: Browsing GKG data.

Step 4: Investigating information from another graph The node representing *Actor A* (purple) in Figure 6.4 is of special interest, and is expanded by the analyst, This reveals information gathered from DBPedia, as shown in Figure 6.5. With this information, more details about this person is available to the analyst to be further explored if needs be.

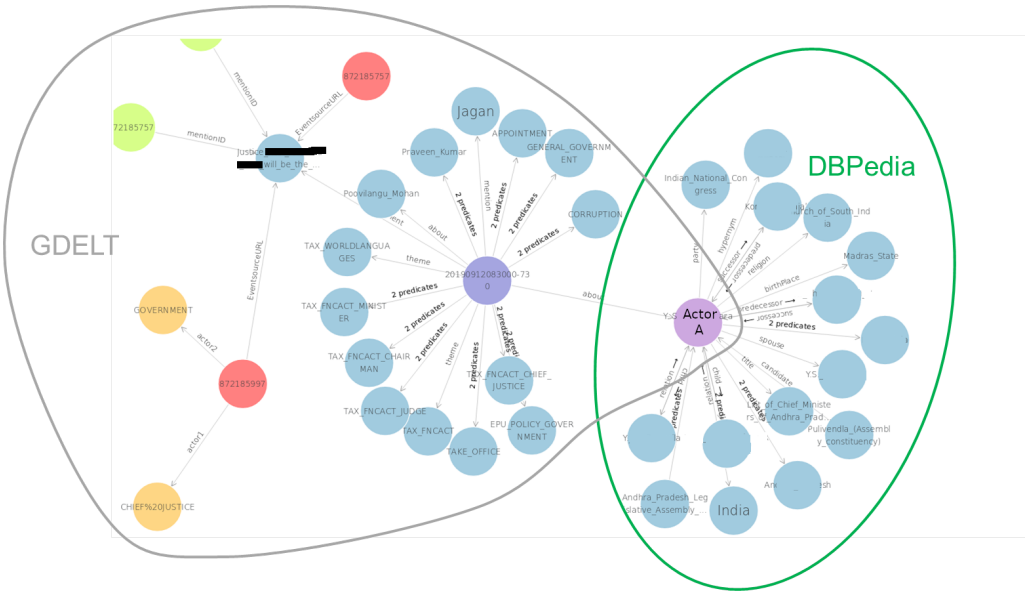


Figure 6.5 Analysis step 4: Expanding the graph with DBPedia information.

Step 5: Following the corruption trail As one of the topics identified as being of special interest at the start of the investigation was corruption, the node representing this topic from GKG was expanded as shown in Figure 6.6. This revealed several new GKG nodes for the analysts to look further into.

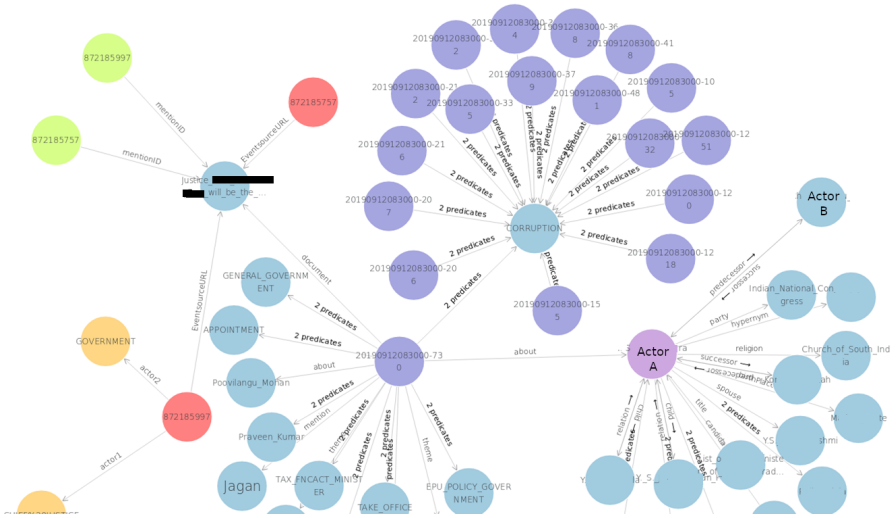


Figure 6.6 Analysis step 5: Considering GKG information related to corruption.

Step 6: Discovering a new link The final step in this investigation was to expand the GKG nodes to see if more interesting information appears. In this case, the GKG node *20190912083000-216* proved the most interesting. As shown in Figure 6.7, this revealed a link to a new individual: *Actor B*. This find provides the analyst team with a new lead on corruption in the area, but their exploration of this lead is a story for another day.

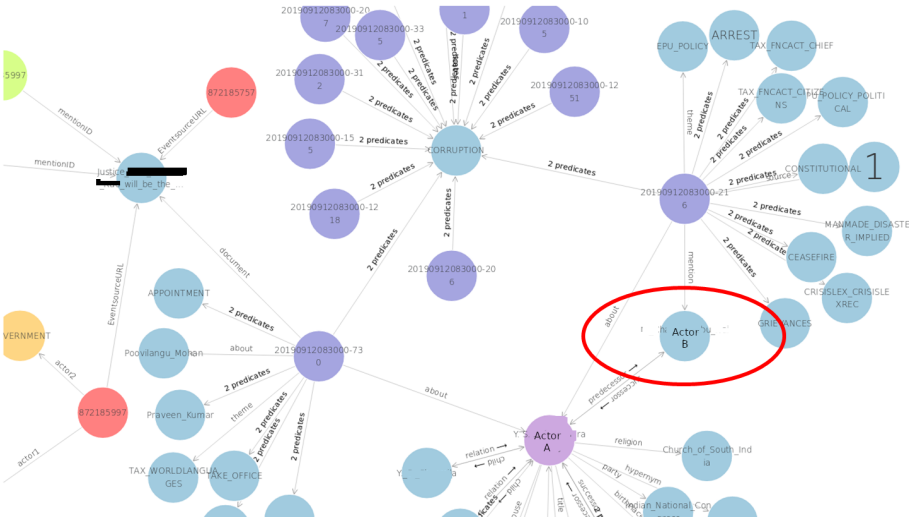


Figure 6.7 Analysis step 6: Discovering new information.

7 Data Reuse: Lineage-based Governance & Access Control

The kind of analysis presented in Chapter 6 opens up a wide range of issues with regards to trust and accountability with respect to data sharing and reuse.

For example, as noted earlier in Chapter 3, the data sharer will likely require that only approved users have access to the shared data, that it is only used in a sanctioned and approved way, and that these restrictions are inherited by derived data products downstream. The former is a requirement that most systems are able to provide, to a varying degree. That is, most systems provide some level of access control, but the granularity that one can form these policies varies highly. The latter is something that is generally less supported, increases complexity substantially, but quickly becomes essential in a big data environment where data will often pass through disparate components for processing and storing.

The potential user, on the other hand, might want to audit where the candidate data originated from (e.g. is the source trustworthy?) and what kind of processing it has undergone along the way. The consumer might want to ask questions such as "does the data satisfy my data quality requirements?" or "is the originator source trustworthy?". Extensive metadata that includes lineage is essential for such auditing.

We note that the issue of trust in data includes a wider range issues, but it is within the sharer and consumer needs outlined above that we scope the exploration. Furthermore, we will leave the data quality side of the trust question aside for now. With that in mind, an overlying governance and policy framework, such as Apache Atlas and Ranger in combination, goes a long way towards making it possible to assert such sought control over a wider ecosystem. Having briefly described the components separately in Chapter 5, we now proceed to explain how these two components played their part in the outlined use case.

First of all, we highlight that the two are designed to tightly co-operate in order to provide the governance and access functionality sought; Atlas acts as a metadata catalogue over artifacts that exist in the numerous heterogeneous systems that exist in the ecosystem, including data lineage and classification tagging of artifacts, which can be referred to in access policies that are managed and enforced by Ranger.

Switching focus over to Ranger, its part in this governance mechanism is that it provides a unified authorization platform where one previously had to maintain separate access control mechanisms for each system type. For the infrastructure described in this report this allowed us to form policy access rules for components, based on security tags assigned to artifacts in Atlas, and have these rules enforced at the component level. For example, as shown in Figure 7.1, we tagged the `dbpedia` Kafka topic with an `Alpha`-classification, added a Ranger tag-based policy that limited all access to `Alpha`-classified objects to a certain user group, which resulted in access to the Kafka topic and any other artifacts downstream of it lineage-wise, being subject to the user belonging to the approved group.

Figure 7.1 shows the lineage capture, as reported to Atlas, of the NiFi flow from Figure 5.2. That is, the individual production components are represented as separate artifacts in Atlas, with associated

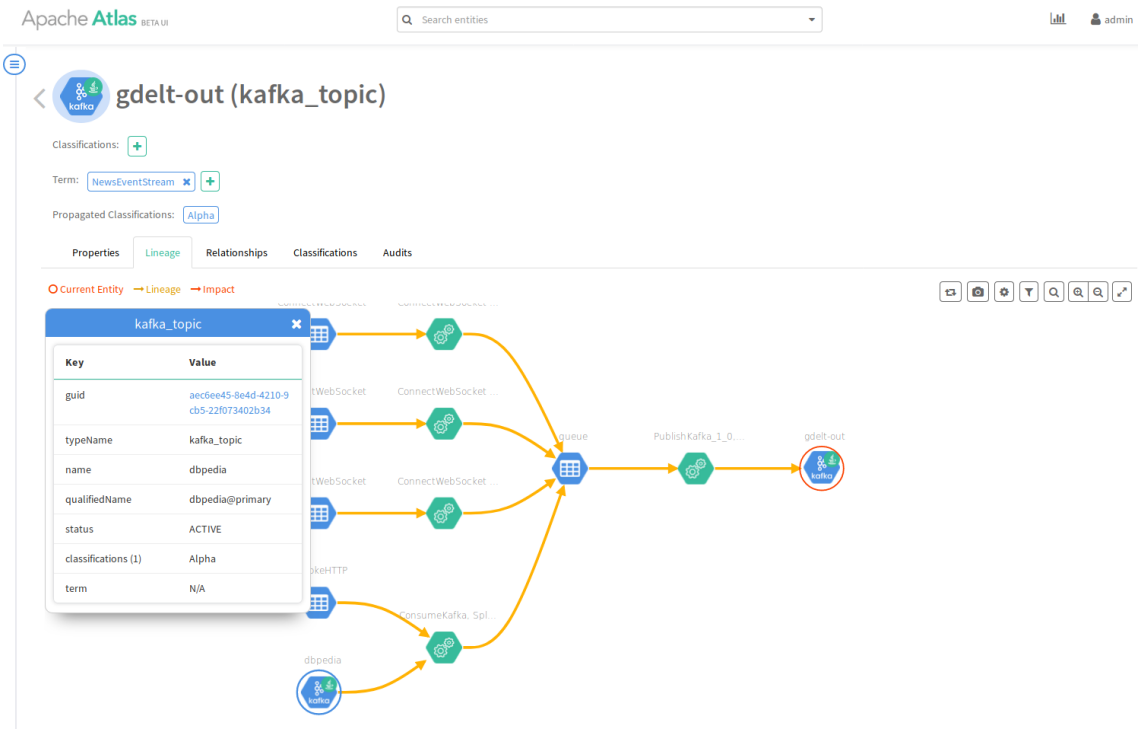


Figure 7.1 NiFi flow reported to Atlas.

metadata, together with edges that capture the data flow. A quick recap of the flow itself might be justified here: the three GDELT flows are merged, together with the DBpedia enrichment, before the aggregated knowledge it output to the `gdelt-out` Kafka topic.

Now, since metadata about both the artifacts and data flow are reported and stored in Atlas, this can be utilized to propagate attributes along lineage lines. Of specific interest in our case is the propagation of security classification tags. Atlas furthermore provides the functionality to propagate tags to downstream artifacts, which Ranger can utilize during policy evaluation.

Policy rules defined in Ranger dictate which user and group credentials map to which classification tags, as well as how access control is to be enforced. In general, Ranger can be set up to enforce access restrictions on any artifact that Atlas has indexed, although enforcement is dependent on the component-level implementation. This means that access control can be defined for files in a distributed file store, Kafka streams, database tables, even down to table columns and rows.

Another feature that makes Ranger's access control interesting, although not explored in the outlined case, is that it allows for artifacts to be tagged with multiple classifications at the same time. In such situations, Ranger's classification tag evaluator applies simple boolean logic to determine authorization approval, meaning that if an artifact is tagged with two different classification tags, the evaluator would require any requestor to be authorized for both tags in order to be allowed to retrieve the object. This, together with lineage-based propagation, affords us a powerful mechanism for complex and fine-grained authorization. In order to keep the running example relatively easy to follow, the complexity of policy rules is kept to a minimum, hence not all features described above are demonstrated.

Returning to the example shown in Figure 7.1, we can see that we have applied classification `Alpha` to the `dbpedia` Kafka stream representation in Atlas. The result being that any downstream artifact, in this case the `gdelt-out` Kafka stream, will also be tagged with the same classification. The Ranger component retrieves tagging information directly from Atlas, as previously described, and applies access controls accordingly. Thus, only users authorized for `Alpha` are able to access the abovementioned two Kafka streams.

Rounding up this section, we have given a short description as to how Atlas and Ranger could provide advanced lineage-based governance and access control in a big data ecosystem.

8 Summary and Discussion

This report set out to explore an open-source technology stack that facilitates extensive secure and auditable information sharing and data reuse in a big data ecosystem, suitable for the military domain.

Specifically, we looked on the following data reuse issues and mitigating technologies:

1. Simplify repurposing and joining of data sets by using the *Linked Data* graph representation scheme, where data elements with unique identifiers are linked together in graphs, represented using RDF and queryable using the RDF graph query language SPARQL.
2. Lineage-based data governance and access control, over a big data ecosystem with many different components, facilitated through the combination of Apache Atlas, a governance and metadata catalogue, and Apache Ranger, a decentralized policy framework.

We were able to furnish a previously explored, loosely coupled big data infrastructure with the abovementioned technologies, and verified that the outlined requirements were successfully satisfied using open-source technologies, i.e. Linked Data, Apache Atlas, and Apache Ranger.

The main point of the exploration is that it consists of loosely coupled components and loosely coupled data. Although the explored infrastructure affords extensive freedom and adaptability when it comes to choice of components and types of data that might exist, it does come with a certain increase in complexity. The complexity lies, of course, in the variety of distributed software components involved; it would undoubtedly be easier to ensure that access and usage restrictions are correctly enforced in a centralized stovepipe system with static data structures. However, such a solution would likely severely hamper the way data can be reused or refined, and would not easily be adaptable for changing needs. That is, either the stovepipe would have to be continuously updated and retrofitted to provide new processing and storage functionality to cater for changing needs, or new processing functionality would need to be done externally which would essentially bypass the security mechanisms afforded by the system. The outlined system, which caters for governance and access control over an ecosystem of loosely coupled system, together with a generic, graph-based data representation, is by design more adaptable and able to handle changing needs, data, components and processing requirements.

One could of course argue that having access and usage restrictions in the first place is the main hindrance for extensive data reuse. However, one highly likely effect of making do without such restrictions is that many data providers would refrain from sharing data, especially within the military domain. Hence, there is a middle ground to be walked; On one hand, we want as much freedom of choice of components as possible in order to utilize the best tools for the problem at hand, while allowing data to flow between them without being hampered. On the other hand, we simultaneously want to ensure that lineage history is being properly recorded and usage restrictions being appropriately applied. We believe that the example infrastructure explored in this report is able to balance this in a meaningful way.

It is worth re-iterating that the component combination put together for this big data infrastructure was chosen for this specific case; the reflections noted in the previous report (Stolpe et al. 2020) still apply, namely that there is no one generic big data infrastructure that is well-suited for all possible

uses, and that the choice of components is highly dependent upon the task at hand. This means that this specific set-up is well-suited for this case, but might not be well-suited for other big data problems. That being said, this experiment adds to the experiences reported on streaming big data infrastructures in general that Apache Nifi and Apache Kafka are robust, well developed components that add value as core components. Furthermore, it is worth noting that although Apache Atlas and Ranger are compatible with a range of big data components (a much larger range of components than demonstrated in this report), there are also many components that are not currently supported.

This report does not provide any direct recommendations or blueprints to follow, since it is very much dependent on the case at hand. Rather, the main contribution of this report is that it provides the reader an example of how today's commonly used big data technologies from the civilian sector can be utilized in the military domain to facilitate data reuse. The results can, however, be seen as supporting evidence of the feasibility of building such a loosely coupled infrastructure, and can be of utility for personnel considering different architectural approaches for dealing with information management in a military setting.

References

Apache Software Foundation (2020*a*), 'Apache Atlas'.
URL: *https://atlas.apache.org*

Apache Software Foundation (2020*b*), 'Apache Kafka Introduction'.
URL: *https://kafka.apache.org/intro*

Apache Software Foundation (2020*c*), 'Apache NiFi Overview'.
URL: *https://nifi.apache.org/docs.html*

Apache Software Foundation (2020*d*), 'Apache Ranger'.
URL: *https://ranger.apache.org*

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R. & Ives, Z. (2007), Dbpedia: A nucleus for a web of open data, *in* 'The semantic web', Springer, pp. 722–735.

Barbieri, D. F., Braga, D., Ceri, S., Valle, E. D. & Grossniklaus, M. (2010), 'Querying RDF streams with C-SPARQL', *ACM SIGMOD Record* **39**(1), 20–26.

Boudemagh, E. & Moise, I. (2017), News Media Coverage of Refugees in 2016: A GDELT Case Study, *in* 'Eleventh International AAAI Conference on Web and Social Media'.

Carroll, J. & Klyne, G. (2004), Resource description framework (RDF): Concepts and abstract syntax, W3C recommendation, W3C. **URL**: https://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

Cloudera (2019), 'Adding NiFi and Kafka to Cloudera Data Platform'.
URL: *https://blog.cloudera.com/adding-nifi-and-kafka-to-cloudera-data-platform/*

Forsvaret (2019), 'Forsvarets fellesoperative doktrine (armed forces joint operational doctrine)'.

Goldstein, J. S. (1992), 'A conflict-cooperation scale for WEIS events data', *Journal of Conflict Resolution* **36**(2), 369–385.

Heath, T. & Bizer, C. (2011), *Linked data: Evolving the web into a global data space*, Morgan & Claypool Publishers.

Herman, I., Melançon, G. & Marshall, M. S. (2000), 'Graph visualization and navigation in information visualization: A survey', *IEEE Transactions on visualization and computer graphics* **6**(1), 24–43.

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutierrez, C., Gayo, J. E. L., Kirrane, S., Neumaier, S., Polleres, A. et al. (2020), 'Knowledge graphs', *arXiv preprint arXiv:2003.02320* .

Johannessen, A. (2019), Infrastruktur for analyser av internasjonale hendelser, FFI-internnotat 19/01643, Norwegian Defence Research Establishment (FFI).

Leetaru, K. & Schrodt, P. A. (2013), Gdelt: Global data on events, location, and tone, *in* 'ISA Annual Convention', Citeseer.

Leetharu, K. (2014), 'Did the Arab Spring Really Spark a Wave of Global Protests?', *Foreign Policy* .

NATO (2008), 'NATO Information Management Policy'. C-M(2007)0118, NATO/PFP UNCLASSIFIED.

Norwegian Digitalisation Agency (2019), 'Veileder for tilgjengeliggjøring av åpne data'. **URL:** *https://doc.difi.no/data/veileder-apne-data/*

Qiao, F., Li, P., Zhang, X., Ding, Z., Cheng, J. & Wang, H. (2017), 'Predicting social unrest events with hidden Markov models using GDELT', *Discrete Dynamics in Nature and Society* **2017**.

Seaborne, A. & Harris, S. (2013), SPARQL 1.1 query language, W3C recommendation, W3C. **URL**: https://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

Stolpe, A., Hansen, B. J. & Halvorsen, J. (2018), Stordatasystemer og deres egenskaper, FFI-rapport 18/01676, Norwegian Defence Research Establishment (FFI).

Stolpe, A., Hansen, B. J., Halvorsen, J. & Opland, E. J. (2020), Experimenting with a big data infrastructure for multimodal stream processing, FFI-rapport 20/00480, Norwegian Defence Research Establishment (FFI).

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E. et al. (2016), 'The FAIR Guiding Principles for scientific data management and stewardship', *Scientific data* **3**.

About FFI

The Norwegian Defence Research Establishment (FFI) was founded 11th of April 1946. It is organised as an administrative agency subordinate to the Ministry of Defence.

FFI's MISSION

FFI is the prime institution responsible for defence related research in Norway. Its principal mission is to carry out research and development to meet the requirements of the Armed Forces. FFI has the role of chief adviser to the political and military leadership. In particular, the institute shall focus on aspects of the development in science and technology that can influence our security policy or defence planning.

FFI's VISION

FFI turns knowledge and ideas into an efficient defence.

FFI's CHARACTERISTICS

Creative, daring, broad-minded and responsible.

Om FFI

Forsvarets forskningsinstitutt ble etablert 11. april 1946. Instituttet er organisert som et forvaltningsorgan med særskilte fullmakter underlagt Forsvarsdepartementet.

FFIs FORMÅL

Forsvarets forskningsinstitutt er Forsvarets sentrale forskningsinstitusjon og har som formål å drive forskning og utvikling for Forsvarets behov. Videre er FFI rådgiver overfor Forsvarets strategiske ledelse. Spesielt skal instituttet følge opp trekk ved vitenskapelig og militærteknisk utvikling som kan påvirke forutsetningene for sikkerhetspolitikken eller forsvarsplanleggingen.

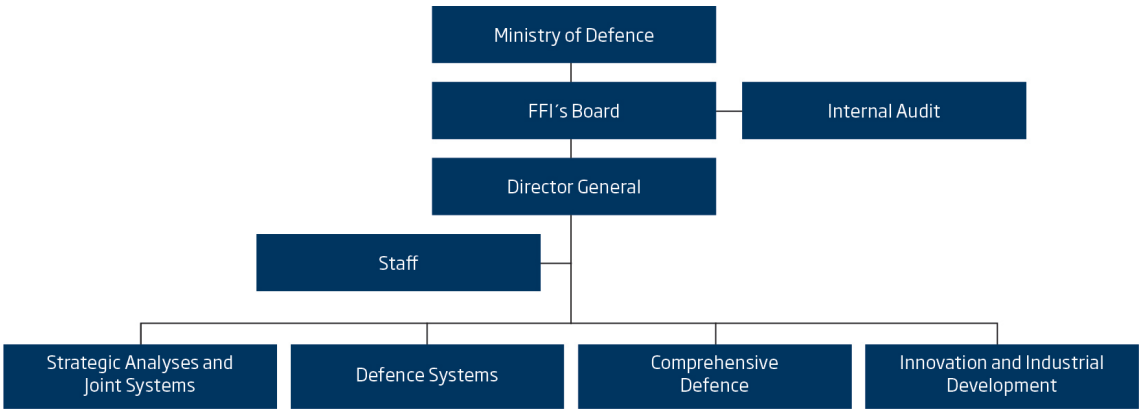
FFIs VISJON

FFI gjør kunnskap og ideer til et effektivt forsvar.

FFIs VERDIER

Skapende, drivende, vidsynt og ansvarlig.

FFI's organisation



Forsvarets forskningsinstitutt
Postboks 25
2027 Kjeller

Besøksadresse:
Instituttveien 20
2007 Kjeller

Telefon: 63 80 70 00
Telefaks: 63 80 71 15
Epost: ffi@ffi.no

Norwegian Defence Research Establishment (FFI)
P.O. Box 25
NO-2027 Kjeller

Office address:
Instituttveien 20
N-2007 Kjeller

Telephone: +47 63 80 70 00
Telefax: +47 63 80 71 15
Email: ffi@ffi.no

FFI Forsvarets forskningsinstitutt